

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2001-94996
(P2001-94996A)

(43) 公開日 平成13年4月6日 (2001.4.6)

(51) Int.Cl.⁷

H 0 4 N 7/32
7/30

識別記号

F I

H 0 4 N 7/137
7/133

テマコード*(参考)

Z 5 C 0 5 9
Z

審査請求 未請求 請求項の数 8 O L (全 24 頁)

(21) 出願番号 特願2000-177646(P2000-177646)

(22) 出願日 平成12年6月13日 (2000.6.13)

(31) 優先権主張番号 特願平11-205258

(32) 優先日 平成11年7月19日 (1999.7.19)

(33) 優先権主張国 日本 (J P)

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 伊木 信弥

東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

(72) 発明者 小林 博

東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

(74) 代理人 100067736

弁理士 小池 晃 (外2名)

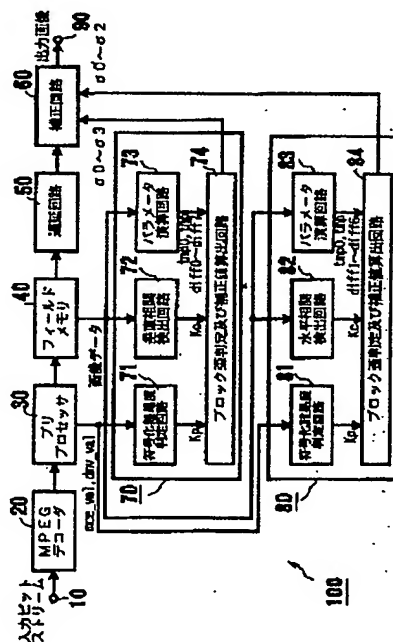
最終頁に続く

(54) 【発明の名称】 ブロック歪低減方法及びブロック歪低減装置

(57) 【要約】

【課題】 処理が簡単で、安定したブロック歪の低減あるいは除去が図れ、さらに、ブロック歪判定の際の誤判定を防止する。

【解決手段】 プリプロセッサ30により入力画像データに動き予測符号化とは独自の動き検出処理を行って得られる動きベクトルと動き補償残差から符号化の難易度を示すパラメータを符号化難易度判定回路74、84により求めるとともに、上記入力画像データからブロック歪判定に必要なパラメータをパラメータ演算回路73、83により演算し、上記符号化の難易度を示すパラメータ及び演算されたパラメータに基づいて、ブロック歪判定及び補正值算出回路74、84により、ブロック歪を判定して上記ブロック歪を低減するための補正值を算出し、上記補正值による補正を補正回路60により上記入力画像データに対して施す。



【特許請求の範囲】

【請求項1】 入力画像データのブロック符号化におけるブロック歪を低減するためのブロック歪低減方法において、

上記入力画像データに動き予測符号化とは独自の動き検出処理を行うことにより、動きベクトルと動き補償残差を算出する動き検出工程と、

上記動き検出工程で検出された動きベクトル及び動き補償残差を符号化の難易度を示すパラメータに変換する符号化難易度検出工程と、

上記入力画像データからブロック歪判定に必要なパラメータを演算するパラメータ演算工程と、

上記符号化難易度検出工程における符号化の難易度を示すパラメータの検出の結果及び上記パラメータ演算工程におけるパラメータの演算結果に基づいてブロック歪を判定するブロック歪判定工程と、

上記ブロック歪を低減するための補正値を算出する補正値算出工程と、

上記ブロック歪判定工程におけるブロック歪の判定結果に応じて、上記補正値算出工程において算出された補正値による補正を上記入力画像データに対して施して出力するブロック歪補正工程とを有することを特徴とするブロック歪低減方法。

【請求項2】 上記動き検出工程では、近隣のブロックにおける動きベクトル同士より動きベクトルの差分を検出し、

上記符号化難易度検出工程では、当該動きベクトル差分及び動き補償残差より符号化の難易度を示すパラメータを算出し、

上記ブロック歪判定工程において当該パラメータによりブロック歪判定を行うことを特徴とする請求項1に記載のブロック歪低減方法。

【請求項3】 上記動き検出工程は、ブロック単位の動きベクトル差分を検出する工程と、画面単位の動きベクトル差分を検出する工程と、ブロック単位の動き補償残差を検出する工程と、画面単位の動き補償残差を検出する工程とを有し、上記符号化難易度検出工程では、上記動き検出工程で検出されたブロック単位の動きベクトル差分、画面単位の動きベクトル差分、ブロック単位の動き補償残差及び画面単位の動き補償残差を符号化の難易度を示すパラメータを算出することを特徴とする請求項2に記載のブロック歪低減方法。

【請求項4】 画像の圧縮率に応じて、上記ブロック歪判定工程におけるブロック歪判定方法及び上記ブロック歪補正工程におけるブロック歪補正方法を制御することを特徴とする請求項1に記載のブロック歪低減方法。

【請求項5】 入力画像データのブロック符号化におけるブロック歪を低減するためのブロック歪低減装置において、

上記入力画像データに動き予測符号化とは独自の動き検出処理を行うことにより、動きベクトルと動き補償残差を算出する動き検出手段と、

上記動き検出手段で検出された動きベクトル及び動き補償残差を符号化の難易度を示すパラメータに変換する符号化難易度検出手段と、

上記入力画像データからブロック歪判定に必要なパラメータを演算するパラメータ演算手段と、

上記符号化難易度検出手段により検出された符号化の難易度を示すパラメータ及び上記パラメータ演算手段により演算されたパラメータに基づいてブロック歪を判定するブロック歪判定手段と、

上記ブロック歪を低減するための補正値を算出する補正値算出手段と、

上記ブロック歪の判定結果に応じた上記補正値による補正を上記入力画像データに対して施して出力するブロック歪補正手段とを有することを特徴とするブロック歪低減装置。

【請求項6】 上記動き検出手段では、近隣のブロックにおける動きベクトル同士より動きベクトルの差分を検出し、

上記符号化難易度検出手段では上記動きベクトル差分及び動き補償残差より符号化の難易度を示すパラメータを算出し、

上記ブロック歪判定手段において上記パラメータに基づいてブロック歪判定を行うことを特徴とする請求項5に記載のブロック歪低減装置。

【請求項7】 上記動き検出手段は、ブロック単位の動きベクトル差分を検出する手段と、画面単位の動きベクトル差分を検出する手段と、ブロック単位の動き補償残差を検出する手段と、画面単位の動き補償残差を検出する手段とを有し、上記符号化難易度検出手段では、上記動き検出手段で検出されたブロック単位の動きベクトル差分、画面単位の動きベクトル差分、ブロック単位の動き補償残差及び画面単位の動き補償残差を符号化の難易度を示すパラメータを算出することを特徴とする請求項6に記載のブロック歪低減装置。

【請求項8】 画像の圧縮率に応じて、上記ブロック歪判定手段及びブロック歪補正手段を制御する有することを特徴とする請求項5に記載のブロック歪低減装置。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】 本発明は、静止画データや動画データ等の入力データをブロック化してDCT符号化等を施すようなブロック符号化におけるブロック歪を低減するためのブロック歪低減方法及びブロック歪低減装置に関する。

【0002】

【従来の技術】 従来、静止画データや動画データ等を効

率よく圧縮符号化するための符号化方式として、ブロックDCT（離散コサイン変換）符号化等のブロック符号化が知られている。

【0003】このようなブロック符号化による画像データ等の圧縮／伸張の際には、ブロック歪（ブロック雑音）が発生することがあり、圧縮率が高くなるほど歪が発生させ易い。このブロック歪は、DCT符号化等がブロック内の閉じた空間で変換を行っており、ブロック境界を越えた相関を考慮していないため、ブロック境界での連続性が保存できず、隣接ブロックとの境界部での再生データ値のずれが雑音として知覚されるものである。画像データをブロック符号化した場合に発生するブロック歪は、一種の規則性を有するため一般のランダム雑音に比べて知覚され易く、画質劣化の大きな要因となっている。

【0004】このブロック歪を低減するために、例えば、「井田、駄竹，“MC-DCT符号化方式におけるノイズ除去フィルタ”、1990年電子情報学会春季全国大会講演論文集、7-35」の文献においては、画像本来の情報であるエッジを保存し、それらのノイズを除去するため、フィルタのオン／オフの決定に量子化ステップサイズを用いたり、処理していく方向を変えて複数回処理を行う技術が開示されている。また、「井澤，“画像のブロック符号化における適応形雑音除去フィルタの特性”，信州大学工学部紀要 第74号、pp.89-100」の文献においては、周辺ブロックまで抜き出してDCT変換を行いノイズ周波数成分を除去する技術が開示されている。

【0005】

【発明が解決しようとする課題】ところで、前者のノイズ除去フィルタをon, offする方法では、処理が簡単な反面、画像の高周波成分が欠落して、解像度が劣化するという欠点がある。

【0006】また、後者の適応形雑音除去フィルタを用いる方法では、解像度が保存されながら効果的なブロック歪の低減が行えるものの、処理が複雑でコストが嵩み、特に民生用機器等に適用するには不适当である。

【0007】また、ブロック歪判定の際の誤判定により、完全にブロック歪が除去できなかったり、エッジを誤補正して擬似エッジを発生させたりする、という問題点がある。

【0008】ブロック符号化方式にて符号化されたビットストリームは、例えばビットレートの異なる機器に対応させるために、例えば8Mbpsから4Mbpsに異なるビットレートのビットストリームにレート変換されることがある。

【0009】このように、ブロック符号化方式にて符号化されたビットストリームをレート変換して再符号化を行う場合には、デコーダで復号された復号画像にブロックノイズが発生していても、そのまま符号化していた。

【0010】このため、再符号化時や、フォーマット変換時のように、動き補償により予測符号化をする動きベクトル検出を行う際には、ブロックノイズに妨害されて動きベクトルの検出精度が低下していた。

【0011】また、ブロックノイズが発生する位置は、ブロックの境界の位置に一意に依存する。そのため、ブロック境界の位置が同じ符号化方式を用いて再エンコードやフォーマットのための符号化を行う場合、ブロックノイズが発生してる画像についてブロックノイズの除去をすることなく符号化すると、再生画像においては、ブロックノイズが強調されてさらに目立ってしまうという問題があった。

【0012】なお、画像データのブロック符号化方式の一例としては、画面内の相関を利用したDCT変換、画面間の相関を利用した動き補償、及び符号列の相関を利用したハフマン符号化を組み合わせたMPEG（Moving pictures experts group）を挙げることができる。

【0013】本発明の目的は、このような従来の実情に鑑みて、DCT符号化方式により圧縮／伸張を行った際に発生するブロック歪を簡易に、かつ、解像度を保存しながら、低減することができるようにしたブロック歪低減方法及びブロック歪低減装置を提供することにある。また、本発明の他の目的は、適応的な処理によりブロック歪判定の誤判定を軽減するとともにブロック歪除去の効果を改善できるようにしたブロック歪低減方法及びブロック歪低減装置を提供することにある。

【0014】

【課題を解決するための手段】本発明は、上述した課題を解決するために、入力画像データのブロック符号化におけるブロック歪を低減するためのブロック歪低減方法において、上記入力画像データに動き予測符号化とは独自の動き検出処理を行うことにより、動きベクトルと動き補償残差を算出する動き検出工程と、上記動き検出工程で検出された動きベクトル及び動き補償残差を符号化の難易度を示すパラメータに変換する符号化難易度検出工程と、上記入力画像データからブロック歪判定に必要なパラメータを演算するパラメータ演算工程と、上記符号化難易度検出工程における符号化の難易度を示すパラメータの検出の結果及び上記パラメータ演算工程におけるパラメータの演算結果に基づいてブロック歪を判定するブロック歪判定工程と、上記ブロック歪を低減するための補正値を算出する補正値算出工程と、上記ブロック歪判定工程におけるブロック歪の判定結果に応じて、上記補正値算出工程において算出された補正値による補正を上記入力画像データに対して施して出力するブロック歪補正工程とを有することを特徴とする。

【0015】また、本発明は、入力画像データのブロック符号化におけるブロック歪を低減するためのブロック歪低減装置において、上記入力画像データに動き予測符号化とは独自の動き検出処理を行うことにより、動きベ

クトルと動き補償残差を算出する動き検出手段と、上記動き検出手段で検出された動きベクトル及び動き補償残差を符号化の難易度を示すパラメータに変換する符号化難易度検出手段と、上記入力画像データからブロック歪判定に必要なパラメータを演算するパラメータ演算手段と、上記符号化難易度検出手段により検出された符号化の難易度を示すパラメータ及び上記パラメータ演算手段により演算されたパラメータに基づいてブロック歪を判定するブロック歪判定手段と、上記ブロック歪を低減するための補正值を算出する補正值算出手段と、上記ブロック歪の判定結果に応じた上記補正值による補正を上記入力画像データに対して施して出力するブロック歪補正手段とを有することを特徴とする。

【0016】

【発明の実施の形態】以下、本発明に係る実施の形態について図面を参照しながら説明する。

【0017】本発明に係るブロック歪低減装置は、例えば図1に示すように構成される。この図1に示すブロック歪低減装置100は、入力端子10と出力端子90の間に縦続接続されたMPEGデコーダ20、プリプロセッサ30、フィールドメモリ40、遅延回路50及び補正回路60と、上記プリプロセッサ30及びフィールドメモリ40にそれぞれ接続された水平方向処理部70及び垂直方向処理部60を備え、上記水平方向処理部70及び垂直方向処理部60により求められた補正值を示す補正データを上記補正回路60に与えることにより、上記MPEGデコーダ20でデコードされた画像データに生ずるブロック歪について、水平方向のブロック歪除去処理と垂直方向のブロック歪除去処理を無相関に行うようになされている。

【0018】このブロック歪低減装置100では、プリプロセッサ30により入力画像データに動き予測符号化とは独自の動き検出処理を行って得られる動きベクトルと動き補償残差から符号化の難易度を示すパラメータを符号化難易度判定回路71、81により求めるとともに、上記入力画像データからブロック歪判定に必要なパラメータをパラメータ演算回路73、83により演算し、上記符号化の難易度を示すパラメータ及び演算されたパラメータに基づいて、ブロック歪判定及び補正值算出回路74、84により、ブロック歪を判定して上記ブロック歪を低減するための補正值を算出し、上記補正值による補正を補正回路60により上記入力画像データに対して施す。

【0019】このブロック歪低減装置100において、MPEGデコーダ20には、入力端子10からの入力ビットストリームとしてMPEG方式の画像データが入力される。このMPEGデコーダ20は、入力された画像データに逆量子化、逆DCT（離散コサイン変換）を施すことで、デコード処理を施す。このとき、MPEGデコーダ2は、複数のマクロブロックからなるDCTブ

ック単位でデコードを施す。

【0020】プリプロセッサ30では、上記MPEGデコーダ20によってデコードされた画像データについて、動きベクトルと動き補償残差を計算し、その値を適応制御のためのパラメータとしてマクロブロック毎のmce_val, dmvm_valに加工して水平方向処理部70及び垂直方向処理部80に供給する。

【0021】ここで、上記プリプロセッサ30は、例えば図2に示すように構成されている。この図2に示したプリプロセッサ30は、入力端子30Aと映像出力端子30Bの間に縦続接続された加算器31、フィールドメモリ32及びフレームシンクロ追越し制御回路33と、上記入力端子30Aとパラメータ出力端子30Cの間に縦続接続された第1の垂直方向位相変換フィルタ34A及び動き補償回路35と、上記フィールドメモリ32と動き補償回路35の間に接続された第2の垂直方向位相変換フィルタ34Bと、上記動き補償回路35と加算器31の間に縦続接続されたアダマール変換回路36、非線型処理回路37及び逆アダマール変換回路38と、上記フィールドメモリ32に接続されたメモリコントローラ39とからなる。

【0022】このプリプロセッサ30では、入力端子30Aを介して入力された画像データについて、フィールド間で動き補償するために、第1の垂直方向位相変換フィルタ34Aを用いて位相変換する。動き補償回路35は、上記第1の垂直方向位相変換フィルタ34Aを介して入力される現フィールドの画像データと第2の垂直方向位相変換フィルタ34Bを介して入力される参照フィールドの画像データとの間で動き補償を行い、得られたフレーム間差分信号をアダマール変換回路36に入力する。アダマール変換によりビデオ信号の特徴を表す8つの成分に分類された差分信号を生成し、この差分信号を非線型処理回路37に通すことで、アダマール変換されたノイズ成分を抽出する。このノイズ成分を逆アダマール変換回路38に通すことで、元の時間軸のノイズ成分信号が得られる。このノイズ成分信号を加算器31に入力し、入力された画像データから時間方向のノイズ成分を除去する。時間方向のノイズ成分を除去した画像データは、フィールドメモリ32に入力され、次のフィールドの参照画像データとして用いられる。この時、書き込みと読み出しの制御を行うために、メモリコントローラ39とフレームシンクロ追越し制御回路33が用いられる。なお、動き補償回路35からは、動き補償残差と動きベクトルの差分値が適応制御のためのパラメータmce_val, dmvm_valとして出力される。

【0023】ここに、mce_val は動き補償残差から算出するパラメータであり、dmvm_val は注目しているブロックの動きベクトルと周囲のブロックの動きベクトルとの差分値である。この時、動き検出を行うブロックの大きさは、MPEG符号化におけるマクロブロックと同じで

なくともよい。

【0024】プリプロセッサ30を経た復号画像データは、フィールドメモリ40に蓄えられ、ブロック歪補正に必要な範囲の画素値のみが上記フィールドメモリ40から読み出されて上記水平方向処理部70及び垂直方向処理部80に供給される。

【0025】上記水平方向処理部70は、上記プリプロセッサ30から適応制御のためのパラメータmce_val, dmv_valが供給される符号化難易度判定回路71と、上記フィールドメモリ40から読み出された画素値が供給される垂直相関検出回路72及びパラメータ演算回路73と、上記各回路71, 72, 73の各出力が供給されるブロック歪判定及び補正值算出回路74とからなる。

【0026】この水平方向処理部70において、符号化難易度判定回路71では、プリプロセッサ30から供給されるパラメータmce_alとdmv_valに基づいてマクロブロック毎の符号化難易度レベルを判定し、現在処理中のマクロブロックの符号化難易度レベルKpを得る。

【0027】ここで、符号化難易度の低い“mobile”と符号化難易度の高い“yashi”という2つの画像の動き補償残差と動きベクトルの差分値のグラフを図3及び図4に示す。各々のグラフから確認できるように、符号化難易度が高く、ブロックノイズが発生し易い“yashi”の方が、動き補償残差も動きベクトルの差分値も、その値が大きい方にまで分布している。すなわち、動き補償残差と動きベクトルの差分値を符号化難易度の指標として用いることができる。

【0028】また、上記水平方向処理部70において、垂直相関検出回路72では、上記フィールドメモリ40から読み出された画素値に基づいて垂直相関係数Kcを算出する。さらに、パラメータ列演算回路73では、上記フィールドメモリ40から読み出された画素値に基づいてパラメータtmp0, tmp, diff0~diff7を算出する。

【0029】そして、ブロック歪判定及び補正值算出回路74では、符号化難易度レベルKpに従ってブロック歪判定のための閾値や係数を変更し、入力されたパラメータtmp0, tmp, diff0~diff7と垂直相関係数Kcからブロックノイズであるか否か、ブロックノイズならばどのようなパターンのブロックノイズであるかの判定を行い、補正值 $\sigma_0 \sim \sigma_3$ を算出して補正回路60に渡す。また、この時、補正中の画素の位置が水平方向ブロック境界と垂直方向ブロック境界の近隣にある場合には、補正值 $\sigma_0 \sim \sigma_3$ を小さめに調整して、補正回路60に渡す。補正回路60では、遅延回路50を経由した画像データに補正值を加算して水平方向の補正処理済みの画像データを得る。ここで、上記ブロック歪判定及び補正值算出回路74によりブロック歪補正を行わないという判定が下された場合には補正值 $\sigma_0 \sim \sigma_3$ には0が値として格納される。

【0030】また、上記垂直方向処理部80は、上記プ

リプロセッサ30から適応制御のためのパラメータmce_val, dmv_valが供給される符号化難易度判定回路81と、上記フィールドメモリ40から読み出された画素値が供給される水平相関検出回路82及びパラメータ演算回路83と、上記各回路81, 82, 83の各出力が供給されるブロック歪判定及び補正值算出回路84とからなる。

【0031】この垂直方向処理部80において、符号化難易度判定回路81では、プリプロセッサ30から供給されるパラメータmce_al, dmv_valに基づいてマクロブロック毎の符号化難易度レベルを判定し、現在処理中のマクロブロックの符号化難易度レベルKpを得る。

【0032】また、水平相関検出回路82では、上記フィールドメモリ40から読み出された画素値に基づいて水平相関係数Kcを算出する。さらに、パラメータ列演算回路83では、上記フィールドメモリ40から読み出された画素値に基づいて上記水平方向処理部70よりも狭い範囲のパラメータtmp0, tmp, diff0~diff7を算出する。

【0033】そして、ブロック歪判定及び補正值算出回路84では、符号化難易度レベルKpに従ってブロック歪判定のための閾値や係数を変更し、入力されたパラメータtmp0, tmp, diff0~diff6と水平相関係数Kcからブロックノイズであるか否か、ブロックノイズならばどのようなパターンのブロックノイズであるかの判定を行い、補正值 $\sigma_0 \sim \sigma_3$ を算出して補正回路60に渡す。また、この時、補正中の画素の位置が水平方向ブロック境界と垂直方向ブロック境界の近隣にある場合には、補正值 $\sigma_0 \sim \sigma_3$ を小さめに調整して、補正回路60に渡す。補正回路60では、遅延回路50を経由した画像データに補正值を加算して垂直方向の補正処理済みの画像データを得る。ここで、上記ブロック歪判定及び補正值算出回路74によりブロック歪補正を行わないという判定が下された場合には補正值 $\sigma_0 \sim \sigma_3$ には0が値として格納される。

【0034】このブロック歪低減装置100により水平方向のブロックノイズ除去処理と垂直方向のブロックノイズ除去処理を無相関に行う場合の画面内適応ブロックノイズ除去処理の手順を図5のフローチャートに示す。

【0035】このブロック歪低減装置100では、フレーム毎にブロックノイズ除去処理を開始する。そして、ブロックノイズ除去処理を開始すると、まず、ステップS1において、水平方向処理部70による水平方向処理のためのマクロブロックのカウンタiをリセットする。

【0036】次のステップS2では、フレーム内でブロックノイズ除去処理する回数をカウントする。水平方向のノイズ除去処理では、

$$MB_NUM = V_MB_NUM \times (H_MB_NUM - 1)$$

とする。

【0037】ここで、V_MB_NUMは垂直方向のマクロプロ

ック数であり、H_MB_NUMは水平方向のマクロブロック数である。所定のブロックノイズ除去を終了した場合は、ステップS 15に進み、垂直方向処理を開始する。

【0038】次のステップS 3では、図6に示す位置のマクロブロック毎の符号化難易度 $Kp_MB[i]$ を計算する。

【0039】次のステップS 4では、符号化難易度の計算ユニットのカウンタ j をリセットする。

【0040】次のステップS 5では、計算ユニットをカウントし、所定の処理が終了した場合は、ステップS 6に進んでカウンタ i をインクリメントする。ここでは、図6に示すように、

UNIT_NUM=2

とする。

【0041】次のステップS 7では、図6に示す計算ユニットにおいて、 $Kp_MB[i]$ と $Kp_MB[i-1]$ から、符号化難易度 Kp を計算する。

【0042】次のステップS 8では、図6に示す計算ユニット内において、垂直関係係数 Kc を計算する。

【0043】次のステップS 9では、 Kp 、 Kc より閾値のDET_THRESHOLD, CORR_THRESHOLDを決定する。

【0044】次のステップS 10では、ライン毎の処理のためのカウンタ k をリセットする。

【0045】次のステップS 11では、ライン毎の処理のためのカウントを行い、所定の処理が終了した場合は、ステップS 12に進んでカウンタ j をインクリメントする。ここで、図6に示すように、フィールド毎に水平方向8ラインの処理を行うため、top_field, bottom_fieldあわせて、LINE_NUM=16である。

【0046】次のステップS 13では、ブロック歪判定及び補正処理を行い、次のステップS 14でカウンタ k をインクリメントして上記ステップS 11に戻る。

【0047】そして、水平方向処理部70による所定のブロックノイズ除去を終了した場合は、上記ステップS 2からステップS 15に進んで垂直方向処理部80による垂直方向処理を行う。上記垂直方向処理部80による垂直方向処理（ステップS 15～S 28）では、計算ユニットなどは図7に示すものを用いて、上述の水平方向処理部70による水平方向処理（ステップS 1～S 14）と同様の処理を行う。

【0048】ここで、上記水平方向処理部70による水平方向処理におけるステップS 3及び上記垂直方向処理部80による垂直方向処理におけるステップS 17で行われるマクロブロック毎の符号化難易度の算出処理について説明する。

【0049】このブロック歪低減装置100では、画像を符号化する際の難易度を示す指標として、上記プリプロセッサ30で得られる次の2つのブロック毎のパラメータを用いている。

【0050】(1) 動きベクトルの差分値 (dmv_val)
周囲のマクロブロックの動きベクトルとの差分値である

ため、動きが大きく複雑であると大きな値となり、動きが小さく単調であると小さな値となる。動きベクトルの差分値が大きいと、符号化効率が低下してブロックノイズが発生し易くなる。周辺のマクロブロックと水平方向ベクトル同士、垂直方向ベクトル同士で差分を取り、その絶対値をパラメータdmv_valとする。

【0051】(2) 動き補償残差 (mce_val)

検出した動きベクトルに従って動き補償をした後の参照画像とのブロック毎の差分値であり、動き補償残差の値が大きい場合、動きが複雑で動き検出が正しく行われていないか、画像のエッジ成分が多いことになる。このような場合、ブロックノイズが発生していることが多い。

【0052】マクロブロック内の画素値毎の差分値の平均をmce_valとする。上記dmv_valとmce_valは必要に応じて画面内で平滑化処理を行った後に所定の方法で正規化する。

【0053】そして、上記水平方向処理部70による水平方向処理では、dmv_valとmce_valからマクロブロックの符号化難易度係数 $Kp_MB[i]$ を

$$Kp_MB[i] = (dmv_val \times WG_DMV + mce_val) / (WG_DMV + WG_MCE)$$

により算出し、また、上記垂直方向処理部80による垂直方向処理では、dmv_valとmce_valからマクロブロックの符号化難易度係数 $Kp_MB[l]$ を

$$Kp_MB[l] = (dmv_val \times WG_DMV + mce_val) / (WG_DMV + WG_MCE)$$

により算出する。ここで、WG_DMVはdmv_valの重みづけ、WG_MCEはmce_valの重み付けである。

【0054】また、上記水平方向処理部70による水平方向処理におけるステップS 7の符号化難易度算出処理では、図8のフローチャートに示すように、ステップS 31において符号化難易度の計算ユニットのカウンタ j の値が0であるか否かを判定しており、その判定結果がYESすなわち $j=0$ の場合は、ブロック境界（マクロブロック内のブロック境界）の処理を行うために、ステップS 32に進んで、

$$Kp = Kp_MB[i]$$

すなわち、 $Kp_MB[i]$ をそのまま符号化難易度係数 Kp として用いる。また、上記ステップS 31における判定結果がNOすなわち $j \neq 0$ でない場合は、図6に示すようにマクロブロック境界（マクロブロック間のブロック境界）の処理を行うために、S 33に進んで、

$$Kp = (Kp_MB[i-1] + Kp_MB[i]) / 2$$

すなわち、 $Kp_MB[i-1]$ と $Kp_MB[i]$ を平均して符号化難易度係数 Kp を求める。

【0055】また、上記垂直方向処理部70による垂直方向処理のステップS 21の符号化難易度算出処理においては、図7に示すように $m=0$ 、 $m=1$ のいずれ場合もマクロブロック境界（マクロブロック間のブロック境界）の処理を行うため、

$$Kp = (Kp_MB[1-1] + Kp_MB[1]) / 2$$

すなわち、 $Kp_B[1-1]$ と $Kp_MB[1]$ を平均して符号化難易度係数 Kp を求める。

【0056】ステップS8の垂直相関係数算出処理では、ブロック境界に沿った方向のエッジの相関の強さを検出する。すなわち、図9のフローチャートに示すように、先ずステップS41においてブロック境界近傍のエッジを抽出し、抽出したエッジに基づいて、次のステップS42で垂直相関係数 Kc を算出する。具体的には、現在処理中のブロック境界の周辺を図10に示すように a 、 b 、 c の領域に分け、各々の領域に含まれるエッジの数 Ea 、 Eb 、 Ec を求め、

$$Kc = Eb / (Ea + Eb + Ec)$$

より垂直相関係数 Kc を求める。図10では、エッジとして抽出された画素を“1”と記し、そうでない画素を“0”と記している。

【0057】上記ステップS41におけるエッジ抽出は、図11に示すような構成のエッジ抽出回路110により行われる。このエッジ抽出回路110は、入力画像データについて、1次元（水平方向）の2次微分信号を抽出するバンドパスフィルタ（BPF）111と、このバンドパスフィルタ111により抽出された2次微分信号が供給される最大値検出回路112と、この最大値検出回路112により検出された最大値が1/2乗算器113を介して供給されるとともに上記バンドパスフィルタ111により抽出された2次微分信号が遅延回路114を介して供給される大小比較回路115からなる。上記遅延回路114は、上記最大値検出回路112及び1/2乗算器113による処理時間に等しい遅延量を有する。

【0058】このエッジ抽出回路110では、入力画像データについて、1次元（水平方向）の2次微分信号をバンドパスフィルタ111により抽出し、この2次微分信号について最大値検出回路112により処理ブロック内で最大値の検出を行い、上記最大値検出回路112により検出した最大値の1/2の値を閾値として大小比較回路115で上記2次微分信号を2値化することにより、エッジ抽出を行う。なお、上記エッジ抽出は、ソフトウェア処理により行うこともできる。

【0059】また、上述の図5に示したフローチャートにおけるステップS9では、垂直相関係数 Kc 及び符号化難易度係数 Kp を用いて、適応的にブロックノイズを除去するために、次のようにして各パラメータを制御する。

【0060】すなわち、垂直相関係数 Kc 及び符号化難易度係数 Kp をもとに、ブロックノイズ除去を行うための閾値 $DET_THRESHOLD$ 、及び、ブロックノイズ除去を行う際の補正値の閾値 $CORR_THRESHOLD$ を変更する。

【0061】 $Kc \& \& Kp \geq HIGH_LEVEL$ の場合には、 $DET_THRESHOLD = DET_THRESHOLD_HIGH$ 、 $CORR_THRESHOLD = CORR_THRESHOLD_HIGH$

とする。また、 $HIGH_LEVEL > Kc \& \& K \geq MID_LEVEL$ の場合には、

$$DET_THRESHOLD = DET_THRESHOLD_MID, CORR_THRESHOLD = CORR_THRESHOLD_MID$$

とする。また、 $MID_LEVEL > Kc \& \& K \geq LOW_LEVEL$ の場合には、

$$DET_THRESHOLD = DET_THRESHOLD_MID, CORR_THRESHOLD = CORR_THRESHOLD_MID$$

とする。さらに、 $Kc \& \& Kp < LOW_LEVEL$ の場合には、

$$DET_THRESHOLD = DET_THRESHOLD_LOW, CORR_THRESHOLD = CORR_THRESHOLD_LOW$$

とする。

【0062】ここで、 $HIGH_LEVEL$ 、 MID_LEVEL 、 LOW_LEVEL は各々補正のレベルを定めた閾値であり、 $DET_THRESHOLD_HIGH$ 、 $DET_THRESHOLD_MID$ 、 $DET_THRESHOLD_LOW$ 、 $CORR_THRESHOLD_HIGH$ 、 $CORR_THRESHOLD_MID$ 、 $CORR_THRESHOLD_LOW$ は各々の補正レベルの補正値における閾値である。

【0063】次に、上述の図5に示したフローチャートにおけるステップS13のブロックノイズ歪判定及び補正処理のアルゴリズムについて説明する。上記ブロックノイズ歪判定及び補正処理のフローチャートを図12に示す。

【0064】図13に水平方向処理のブロックノイズ除去の範囲を示し、また、図14に垂直方向処理のブロックノイズ除去の範囲を示す。

【0065】ブロックノイズ歪判定及び補正処理では、先ずステップS51において、次のようにして各パラメータを算出する。

【0066】

$$diff0 = b - a$$

$$diff1 = c - b$$

$$diff2 = d - c$$

$$diff3 = e - d$$

$$tmp0 = f - e$$

$$diff4 = g - f$$

$$diff5 = h - g$$

$$diff6 = i - h$$

$$diff7 = j - i$$

$$tmp = (|diff0| + |diff1| + |diff2| + |diff3| + |diff4| + |diff5| + |diff6| + |diff7|) / 8$$

次のステップS52では、ブロックノイズ除去を行うための閾値を $DET_THRESHOLD$ として、

$$|tmp0| > tmp \& \& |tmp0| < DET_THRESHOLD$$

すなわち、ブロック境界に段差があり（ブロック境界差分 $(|tmp0|) >$ 周辺画素のアクティビティ (tmp) ）、かつ、ブロック境界にエッジがない（ブロック境界差分 $(|tmp0|) < DET_THRESHOLD$ ）状態であるか否かを判定する。

【0067】そして、このステップS52における判定結果がYESである場合にはステップS53に進み、N

0である場合にはステップS60に進んで、原信号をそのまま出力する。

【0068】ステップS53では、次の3つの条件

(1)～(4)をすべて満たすか否かを判定する。

【0069】条件(1) : $\text{diff}0 \sim \text{diff}7 > 0$ 又は $\text{diff}0 \sim \text{diff}7 < 0$

すなわち、ブロック境界を除いた補正範囲の輝度値の変化が単調減少か単調増加である。

条件2 : $\text{tmp}0 \times (\text{diff}0 \sim \text{diff}7) < 0$

すなわち、ブロック境界での輝度値の変化が補正範囲の他の輝度値の変化と逆である。

条件(3) : $Kp \geq \text{DIFF_THRESHOLD}$

すなわち、処理中のブロック境界の符号化難易度が十分高い。

条件(4) : $Kc \geq \text{DIFF_THRESHOLD}$

すなわち、処理中のブロック境界の垂直あるいは水平相関が十分高い。

【0070】そして、このステップS53における判定結果がNOである場合にはステップS54に進み、YESである場合にはステップS61に進む。

【0071】ステップS54では、

$|\text{tmp}0| \geq |\text{diff}3| \& \& |\text{tmp}0| \geq |\text{diff}4|$

すなわち、ブロック境界の両端にエッジがない状態であるか否かを判定する。

【0072】そして、このステップS54における判定結果がYESである場合、すなわち、ブロックノイズ除去を行うための閾値をDET_THRESHOLDとして、ステップS52及びステップS54で次の条件(11)～(13)をすべて満たす場合に、

条件(11) : ブロック境界差分($|\text{tmp}0|$) > 周辺画素のアクティビティ(tmp) すなわち、ブロック境界に段差がある。

条件(12) : ブロック境界差分($|\text{tmp}0|$) < DET_THRESHOLD すなわち、ブロック境界にエッジがない。

条件(13) : ブロック境界差分($|\text{tmp}0|$) \geq 境界隣接差分($|\text{diff}3| \& |\text{diff}4|$) すなわち、ブロック境界の両端にエッジがない。

ステップS55に進んで特殊処理1を行い、NOである場合にはステップS64に進んで特殊処理2を行う。

【0073】そして、上記条件11～13をすべて満たす場合、ブロックノイズ除去の補正値を決めるための閾値をCORR_THRESHOLDとして次のように補正値 $\sigma 0$ を決める。

【0074】すなわち、ブロック境界差分($|\text{tmp}0|$) \geq CORR_THRESHOLD又は垂直方向ブロック境界と水平方向ブロック境界の交差位置近傍では、

補正値 $\sigma 0 = (|\text{tmp}0| - |\text{step}|) / 4$

とする。また、ブロック境界差分($|\text{tmp}0|$) < CORR_THRESHOLDの場合には、

補正値 $\sigma 0 = (|\text{tmp}0| - |\text{step}|) / 2$

とする。

【0075】ここに、補正後の段差stepは、

$\text{step} = (\text{diff}3 + \text{diff}4) / 2$

である。

【0076】ここで、ブロック境界での段差がある程度大きい場合には、それがブロックノイズではなく、本来のエッジである可能性が高いために、このようにして補正値を小さく抑える。また、当該ブロック境界が垂直方向ブロック境界と水平方向ブロック境界の交差位置近傍にある場合には、互いの補正により破綻が生じる可能性があるために、後述する特殊処理1や特殊処理2で補正値を調整する。次のステップS56ではブロック境界の相関係数Kc及び符号化難易度係数Kpをブロックノイズ補正値 $\sigma 0$ に乗ずる。

【0077】 $\sigma 0 = \sigma 0 \times Kc \times Kp$

さらに次のステップS57では隣り合う補正範囲同士の繋ぎ目を滑らかにするために、図15のように補正範囲の外側に行くに従って補正値が小さくなるように補正値 $\sigma 1$, $\sigma 2$, $\sigma 3$ に重み付けを行う。

【0078】

補正値 $\sigma 1 = \sigma 0 / 2$

補正値 $\sigma 2 = \sigma 0 / 4$

補正値 $\sigma 3 = \sigma 0 / 8$

なお、垂直方向処理においては、補正値 $\sigma 3$ はない。

【0079】次のステップS57では、重み付けされた補正値 $\sigma 1$, $\sigma 2$, $\sigma 3$ を原信号に加算し、補正済みの画像データを得る。

【0080】そして、次のステップS58では、ブロック歪処理済みの画像データを出力する。

【0081】ブロックノイズの特性によっては、上記のアルゴリズムによってノイズ除去ができない場合があるため、各々の状況に応じて適応的に処理を行うアルゴリズムを追加することにより、より効果的なブロックノイズの除去を行う。

【0082】ここで、図17に示すようなブロックノイズ除去処理では破綻を生じる可能性が高いために、使用に際しては注意が必要となる。そこで、マクロブロック毎の符号化難易度及びマクロブロック境界の垂直相関度に応じて適応的にこれらのブロックノイズ除去処理を行うようにすることで、破綻を生じ難くする。

【0083】図16に補正が十分でない場合の例を示す。この例では、 $\text{diff}0 \sim \text{diff}7$ まですべてが負の値を取り、tmp0のみが正の値を取る。すなわち、画素a～eでは輝度値が単調減少していき、ブロック境界の所でのみ輝度値が増加し、再び画素f～jで輝度値が単調減少している。このような場合、通常の補正を行うと図16の例のように、補正が十分でないために、ブロック境界に段差が残ってしまい、ブロックノイズを除去し切れない場合がある。

【0084】ここで、図16の例のように画素a～eで

は輝度値が単調減少していて、ブロック境界の所でのみ輝度値が増加し、再び画素 $f \sim j$ で輝度値が単調減少している場合には、原画像ではブロック境界の所でも輝度値が単調減少していたと考え、図 17 のような補正を行うことによりブロックノイズを除去できる。

【0085】このような場合を検出するために上記ステップ S 53 で上述の条件 1, 2, 3, 4 をすべて満たすか否かを判定する。そして、上記ステップ S 53 における判定結果が YES の場合には、ステップ S 61 に進んで、

ブロック境界差分 ($|tmp0|$) \geq CORR_THRESHOLD

又は垂直方向ブロック境界と水平方向ブロック境界の交差位置近傍であるかを判定する。

【0086】このステップ S 61 の判定結果が YES、すなわち、

ブロック境界差分 ($|tmp0|$) \geq CORR_THRESHOLD

又は垂直方向ブロック境界と水平方向ブロック境界の交差位置近傍である場合には、ステップ S 62 に進んで、

補正值 $\sigma 0 = (|tmp0| + |step|) / 4$

とする。また、上記ステップ S 61 の判定結果が NO、すなわち、

ブロック境界差分 ($|tmp0|$) $<$ CORR_THRESHOLD

である場合には、ステップ S 63 に進んで、

補正值 $\sigma 0 = (|tmp0| + |step|) / 2$

とする。

【0087】ここに、補正後の段差 $step$ は、

$step = (|diff3| + |diff4|) / 2$

である。

【0088】そして、この場合も、ステップ S 56 でブロック境界の相関係数 Kc 及び符号化難易度係数 Kp を乗じたブロックノイズ補正值 $\sigma 0$ にステップ S 57 で重み付けを行う。

【0089】

補正值 $\sigma 1 = \sigma 0 / 2$

補正值 $\sigma 2 = \sigma 0 / 4$

補正值 $\sigma 3 = \sigma 0 / 8$

なお、垂直方向処理においては、補正值 $\sigma 3$ はない。

【0090】ここで、上述の図 12 のフローチャートにおけるステップ S 55 では、図 18 のフローチャートに示すような特殊処理 1 を行う。

【0091】この特殊処理 1 は、例えば図 19 に示すような補正の結果、新たにエッジが生じる場合に対応するための特殊処理である。図 19 の例では、 $|diff3|$ と $|diff4|$ の値が非常に近く、なおかつ $|tmp0|$ が比較的大きいために、 $step$ の値が $\sigma 0$ に比べて極端に小さくなってしまっている。このため、補正を行った結果、ブロック境界の近傍画素とそれに近接する画素の間で新たにエッジを生じてしまい、これがブロックノイズとして知覚されてしまう。

【0092】そこで、特殊処理 1 では、先ずステップ S

71 でブロック境界差分 ($|tmp0|$) が閾値 (CORR_THRESHOLD) よりも大きいかな否かを判定する。そして、このステップ S 71 における判定結果が YES、すなわち

$|tmp0| \geq$ CORR_THRESHOLD

の場合にはステップ S 72 に進んで

補正值 $\sigma 0 = (|tmp0| - |step|) / 4$

とする。また、上記ステップ S 71 における判定結果が NO、すなわち

$|tmp0| <$ CORR_THRESHOLD

の場合にはステップ S 73 に進む。

【0093】このステップ S 73 では、補正後の段差 $|step|$ と補正值 $\sigma 0$ の比 CORR_RATIO を $CORR_RATIO = |step| / \sigma 0$ として、

$|step| < \sigma 0 \times CORR_RATIO$

又は垂直方向ブロック境界と水平方向ブロック境界の交差位置近傍であるかを判定する。

【0094】このステップ S 73 の判定結果が YES、すなわち、

$|step| < \sigma 0 \times CORR_RATIO$

すなわち、補正後の段差 $step$ が補正值 $\sigma 0$ に比べて一定の比よりも小さい場合、又は垂直方向ブロック境界と水平方向ブロック境界の交差位置近傍である場合には、ステップ S 74 に進んで、

補正值 $\sigma 0 = (|tmp0| - |step|) / 4$

とする。また、上記ステップ S 73 の判定結果が NO、すなわち、

ブロック境界差分 ($|tmp0|$) $<$ CORR_THRESHOLD

である場合には、ステップ S 75 に進んで、

補正值 $\sigma 0 = (|tmp0| - |step|) / 2$

とする。

【0095】この場合、図 20 における周辺画素の補正值 $\sigma 1$, $\sigma 2$, $\sigma 3$ は以下のようにして求める。

【0096】

補正值 $\sigma 1 = \sigma 0' / 2$

補正值 $\sigma 2 = \sigma 0' / 4$

補正值 $\sigma 3 = \sigma 0' / 8$

このようにすることで、図 20 に示すように、補正後の段差 $|step|$ が小さいことに起因する新たなエッジの発生を未然に防ぎ、より破綻の少ないブロックノイズ除去を行うことができる。

【0097】さらに、 $|diff3| > |tmp0|$ であるためにブロックノイズとして検出される条件を満たしていないが、ブロックノイズとして知覚できる場合の例を図 21 に示す。

【0098】このような場合を検出するために、このブロック歪低減装置 100 では、通常のブロックノイズ除去のための判定の他に、上述の図 12 のフローチャートにおけるステップ S 64 で図 22 のフローチャートに示すような特殊処理 2 を行う。

【0099】特殊処理 2 では、先ずステップ S 81 に

いて、 $\text{diff3} \times \text{tmp0} < 0 \ \&\& \text{diff4} \times \text{tmp0} \geq 0 \ \&\& \text{diff2} \times \text{diffi3} \geq 0$ すなわち、ブロック境界に近接する画素 e と画素 f の間にエッジがあるか否かを判定する。

【0100】このステップS81における判定結果がYES すなわちブロック境界に近接する画素 e と画素 f の間にエッジがある場合にはステップS82に進み、NO の場合にはステップS86に進む。

【0101】そして、ステップS82では、ブロック境界差分 ($|\text{tmp0}|$) $\geq \text{CORR_THRESHOLD}$ 又は垂直方向ブロック境界と水平方向ブロック境界の交差位置近傍であるかを判定する。

【0102】このステップS82における判定結果がYES、すなわち

ブロック境界差分 ($|\text{tmp0}|$) $\geq \text{CORR_THRESHOLD}$ 又は垂直方向ブロック境界と水平方向ブロック境界の交差位置近傍である場合には、ステップS83に進んで、補正值 $\sigma 0 = (|\text{tmp0}| + |\text{diff4}|) / 4$

とする。また、上記ステップS82の判定結果がNO、すなわち、

ブロック境界差分 ($|\text{tmp0}|$) $< \text{CORR_THRESHOLD}$ である場合には、ステップS84に進んで、補正值 $\sigma 0 = (|\text{tmp0}| + |\text{diff4}|) / 2$ とする。

【0103】そして、ステップS85において、補正值 $\sigma 1$ 、 $\sigma 2$ は

補正值 $\sigma 1 = \text{diff2} / 2$

補正值 $\sigma 2 = \text{diff4} / 2$

として、ステップS92に進む。

【0104】また、上記ステップS86では、 $\text{diff4} \times \text{tmp0} < 0 \ \&\& \text{diff3} \times \text{tmp0} \geq 0 \ \&\& \text{diff4} \times \text{diff5} \geq 0$

すなわち、画素 f と画素 g の間にエッジがあるか否かを判定する。

【0105】このステップS86における判定結果がYES すなわち画素 f と画素 g の間にエッジがある場合には、ステップS87に進み、NO の場合にはステップS94に進んで、原信号をそのまま出力して、この特殊処理2を終了する。

【0106】そして、ステップS87では、ブロック境界差分 ($|\text{tmp0}|$) $\geq \text{CORR_THRESHOLD}$ 又は垂直方向ブロック境界と水平方向ブロック境界の交差位置近傍であるかを判定する。

【0107】このステップS87の判定結果がYES、すなわち、

ブロック境界差分 ($|\text{tmp0}|$) $\geq \text{CORR_THRESHOLD}$

又は垂直方向ブロック境界と水平方向ブロック境界の交差位置近傍である場合には、ステップS88に進んで、補正值 $\sigma 0 = (|\text{tmp0}| + |\text{diff4}|) / 4$

とする。また、上記ステップS87の判定結果がNO、すなわち、

ブロック境界差分 ($|\text{tmp0}|$) $< \text{CORR_THRESHOLD}$

である場合には、ステップS89に進んで、

補正值 $\sigma 0 = (|\text{tmp0}| + |\text{diff4}|) / 2$

とする。

【0108】次のステップS90ではブロック境界の相関係数Kc及び符号化難易度係数Kpをブロックノイズ補正值 $\sigma 0$ に乗ずる。

【0109】 $\sigma 0 = \sigma 0 \times Kc \times Kp$

そして、ステップS91において、補正值 $\sigma 1$ 、 $\sigma 2$ は補正值 $\sigma 1 = \text{diff3} / 2$

補正值 $\sigma 2 = \text{diff5} / 2$

とする。

【0110】次のステップS92では、重み付けされた補正值 $\sigma 1$ 、 $\sigma 2$ を原信号に加算し、補正済みの画像データを得る。

【0111】そして、次のステップS93では、ブロック歪処理済みの画像データを出力して、この特殊処理2を終了する。

【0112】このような判定処理を行うことで、ブロック境界に近接する画素上にエッジがある場合にもブロックノイズとして検出が可能となる。

【0113】上述の図1に示したブロック歪低減装置100では、水平、垂直方向のブロック境界に対して、無相関にブロック歪判定処理、補正処理を行うようにしたが、図23に示すような構成により、水平、垂直方向のブロック境界に対して、先に水平／垂直のブロック歪判定処理及び補正処理を行い、その後に垂直／水平のブロック歪判定処理及び補正処理を行うようにすることもできる。

【0114】この図23に示すブロック歪低減装置200は、入力端子210と出力端子290の間に縦続接続されたMPEGデコーダ220、プリプロセッサ230、第1のフィールドメモリ241、第1の遅延回路271、第1の補正回路281、第2のフィールドメモリ242、第1の遅延回路272、及び第2の補正回路282と、上記プリプロセッサ230及び第1のフィールドメモリ241に接続された垂直方向処理部250と、上記プリプロセッサ230及び第2のフィールドメモリ242に接続された水平方向処理部260を備える。

【0115】このブロック歪低減装置200では、上記垂直方向処理部250により水平／垂直のブロック歪判定処理を行って得られる補正值 $\sigma 0 \sim \sigma 2$ を用いて第1の補正回路281により補正処理を行い、その後に上記水平方向処理部260により垂直／水平のブロック歪判定処理を行って得られる補正值 $\sigma 0 \sim \sigma 3$ を用いて第2の補正回路282により補正処理を行う。

【0116】この場合、後段のブロック歪判定処理及び補正処理においてのみ、境界の交差点における補正值の調整を行う以外は、上述の水平及び垂直方向で無相関にブロック歪判定処理及び補正処理を行う場合と全く同じ

処理になる。

【0117】また、処理の順序が逆になった場合でも、処理内容には変化はないが、視覚効果を考える場合、より影響の強い水平方向処理を後に行う方が有効である。

【0118】これまでの説明では、マクロブロック単位で符号化難易度に応じて適応的にブロックノイズ除去処理を行うようにしたが、画面単位とマクロブロック単位で符号化難易度を算出するようにしてもよい。

【0119】以下、画面単位とマクロブロック単位で符号化難易度を算出して適用する例について、図24のフローチャートを参照して説明する。

【0120】このブロック歪低減装置100では、フィールド毎にブロックノイズ除去処理を開始する。そして、ブロックノイズ除去処理を開始すると、まず、ステップS100において、水平方向処理部70による水平方向処理のためのマクロブロックのカウント*i*をリセットする。

【0121】次のステップS101では、フレーム内でブロックノイズ除去処理する回数をカウントする。水平方向のノイズ除去処理では、

$$MB_NUM = V_MB_NUM \times (H_MB_NUM - 1)$$

とする。

【0122】ここで、*V_MB_NUM*は垂直方向のマクロブロック数であり、*H_MB_NUM*は水平方向のマクロブロック数である。所定のブロックノイズ除去を終了した場合は、ステップS115に進み、垂直方向処理を開始する。

【0123】次のステップS102では、図6に示す位置のマクロブロック毎の符号化難易度*Kp_MB[i]*を計算する。

【0124】次のステップS103では、フィールド毎の符号化難易度*Kp_Fi*を計算する。

【0125】

【数1】

$$Kp_Fi = 1 / MB_NUM \times \sum_{i=0}^{MB_NUM-1} Kp_MB[i]$$

【0126】次のステップS104では、符号化難易度の計算ユニットのカウント*j*をリセットする。

【0127】次のステップS105では、計算ユニットをカウントし、所定の処理が終了した場合は、ステップS106に進んでカウント*i*をインクリメントする。ここでは、図6に示すように、

$$UNIT_NUM = 2$$

とする。

【0128】次のステップS107では、図6に示す計算ユニットにおいて、*Kp_MB[i]*と*Kp_MB[i-1]*及び*Kp_Fi*から、符号化難易度*Kp*を計算する。

【0129】次のステップS108では、図6に示す計算ユニット内において、垂直相関係数*Kc*を計算する。

【0130】次のステップS109では、*Kp*、*Kc*より閾値の*DET_THRESHOLD*、*CORR_THRESHOLD*を決定する。

【0131】次のステップS110では、ライン毎の処理のためのカウンタ*k*をリセットする。

【0132】次のステップS111では、ライン毎の処理のためのカウントを行い、所定の処理が終了した場合は、ステップS112に進んでカウンタ*j*をインクリメントする。ここで、図6に示すように、フィールド毎に垂直方向8ラインの処理を行うため、

$$LINE_NUM = 8$$

である。

【0133】次のステップS113では、ブロック歪判定及び補正処理を行い、次のステップS114でカウンタ*k*をインクリメントして上記ステップS111に戻る。

【0134】そして、水平方向処理部70による所定のブロックノイズ除去を終了した場合は、上記ステップS101からステップS115に進んで垂直方向処理部80による垂直方向処理を行う。上記垂直方向処理部80による垂直方向処理（ステップS115～S129）では、計算ユニットなどは図7に示すものを用いて、上述の水平方向処理部70による水平方向処理（ステップS100～S114）と同様の処理を行う。

【0135】ここで、フィールド毎の符号化難易度*Kp_Fi*の算出過程において、静止画検出、シーンチェンジ検出、時間軸フィルタ処理を施す例について、図25、図26及び図27を参照して説明する。

【0136】すなわち、この例において、ステップS201では、入力画像が静止画であるか否かを判定する。例えば符号化難易度*Kp_Fi*の値が複数フィールド間、同じ値あるいは近い値で連続する場合に静止画であると判定する。

【0137】上記ステップS201において、入力画像が静止画であると判定した場合には、ステップS202に進み、符号化難易度減衰値*d*に例えば定数2を加算する。また、入力画像が静止画でないと判定した場合には、ステップS203に進み、符号化難易度減衰値*d*に定数0を代入する。

【0138】図26のタイミングチャートに示す例では、フィールド毎の符号化難易度*Kp_Fi*の値が16フィールド連続した場合に静止画であると判定しており、静止画がフィールド番号[n+1]～フィールド番号[n+23]まで連続したとすると、フィールド番号[n+16]から符号化難易度減衰値*d*が定数2ずつカウントアップされ、入力画像が静止画でないと判定されるフィールド番号[n+24]で符号化難易度減衰値*d*が再び0となる。

【0139】次のステップS204では、フィールド毎の符号化難易度*Kp_Fi*から符号化難易度減衰値*d*を減算する。

【0140】そして、ステップS205、S206では、上記ステップS204での減算結果*Kp_Fi_stl*が負にならないように制限する。すなわち、ステップS20

5ではステップS204での減算結果 $Kp_Fi_stl[0]$ が0よりも小さいか否かを判定し、減算結果 $Kp_Fi_stl[0]$ が0よりも小さい場合にはステップS206で $Kp_Fi_stl[0]=0$ とする。

【0141】次のステップS207では、シーンチェンジ点か否かの判定を行う。シーンチェンジ点の判定結果を示すフラグ $xsc[0]$ は、図27のタイミングチャートに示すように、シーンチェンジ点の後1フィールド間Lowレベルになる。そして、フラグ $xsc[0]$ は、1フィールド前のフラグ $xsc[F]$ 及び2フィールド前のフラグ $xsc[2F]$ との論理積演算により、時間軸上で伸張されて、シーンチェンジ点の後3フィールド間をLowレベルで示すシーンチェンジ信号 $xsc[0]\&xsc[F]\&xsc[2F]$ とされる。

【0142】上記シーンチェンジ信号 $xsc[0]\&xsc[F]\&xsc[2F]$ がLowレベルのとき、すなわち、上記ステップS207における判定結果がYESの場合にシーンチェンジ処理を行い、ステップS208で1フィールド前のシーンチェンジ符号化難易度 $Kp_Fi_sc[F]$ を保持する。一方、上記シーンチェンジ信号 $xsc[0]\&xsc[F]\&xsc[2F]$ がHighレベルのとき、すなわち、上記ステップS207における判定結果がNOの場合には、ステップS209に進んで静止画処理を行った符号化難易度 $Kp_Fi_stl[0]$ をシーンチェンジ処理後の符号化難易度 $Kp_Fi_sc[0]$ とする。

【0143】そして、次のステップS210では、上記シーンチェンジ符号化難易度 $Kp_Fi_sc[0]$ 、 $Kp_Fi_sc[F]$ を用いて、

$Kp_Fi_tfil[0] = (Kp_Fi_sc[0] + Kp_Fi_sc[F]) / 2$
とする時間軸方向のフィルタリング処理を施す。

【0144】以上の説明では、マクロブロック毎に算出した符号化難易度と1フィールド毎に算出した符号化難易度とを組み合わせ適用したが、画面単位だけで符号化難易度を算出して適用してもよい。

【0145】なお、画面単位とは、1フィールドに限らず、1フレーム、複数フレームであってもよい。

【0146】また、マクロブロックの大きさは、1画素×1画素から画面内の任意の大きさをとることができる。

【0147】次に、マクロブロック毎に算出した符号化難易度と1フィールド毎に算出した符号化難易度とを組み合わせ適用する場合において、画像の圧縮率すなわち画像伝送速度に応じてブロック歪判定のための閾値を制御する例について、図28及び図29を参照して説明する。

【0148】すなわち、マクロブロック毎に算出した符号化難易度 Kp_MB と1フィールド毎に算出した符号化難易度 Kp_Fi とを組み合わせ適用する場合には、フィールド毎に算出した符号化難易度 Kp_Fi を閾値として図28の(A)、(B)に示すテーブルに従って $CORR_THR_Fi$

i、 DET_THR_Fi を決定するとともに、マクロブロック毎に算出した符号化難易度 Kp_MB を閾値として図29の(A)、(B)のテーブルに従って重み付け係数 $a(Kp_MB)$ を決定し、ブロック歪判定処理及び補正処理において使用するパラメータ $CORR_THRESHOLD$ 、 $DET_THRESHOLD$ を、
 $CORR_THRESHOLD = a(Kp_MB) \times CORR_THR_Fi$
 $DET_THRESHOLD = a(Kp_MB) \times DET_THR_Fi$
 にてブロック毎に算出する。

【0149】図28には、符号化難易度 Kp_Fi から $CORR_THR_Fi$ 、 DET_THR_Fi を決定するのに使用するテーブルであって、画像伝送速度が4Mbpsのときの例を(A)に示し、画像伝送速度が8Mbpsのときの例を(B)に示してある。

【0150】すなわち、例えば画像伝送速度が4Mbpsで算出された符号化難易度 Kp_Fi が例えば $8 \leq Kp_Fi < 12$ のときには、図28の(A)のテーブルにより $CORR_THR_Fi = 40$ 、 $DET_THR_Fi = 56$

という値が設定される。また、例えば画像伝送速度が8Mbpsで算出された符号化難易度 Kp_Fi が例えば $4 \leq Kp_Fi < 6$ のときには、図28の(B)のテーブルにより $CORR_THR_Fi = 24$ 、 $DET_THR_Fi = 32$ という値が設定される。

【0151】また、図29には、符号化難易度 Kp_MB から重み付け係数 $a(Kp_MB)$ を決定するのに使用するテーブルであって、画像伝送速度が4Mbpsのときの例を(A)に示し、画像伝送速度が8Mbpsのときの例を(B)に示してある。

【0152】すなわち、例えば画像伝送速度が4Mbpsで算出された符号化難易度 Kp_MB が例えば $10 \leq Kp_MB < 14$ のときには、図29の(A)のテーブルにより $a(Kp_MB) = 1.0$

という値が設定される。また、例えば画像伝送速度が8Mbpsで算出された符号化難易度 Kp_MB が例えば $8 \leq Kp_MB < 12$ のときには、図29の(B)のテーブルにより

$a(Kp_MB) = 0.8$

という値が設定される。

【0153】したがって、例えば画像伝送速度が4Mbpsで符号化難易度 Kp_Fi が $8 \leq Kp_Fi < 12$ 、符号化難易度 Kp_MB が $10 \leq Kp_MB < 14$ であるときには、
 $CORR_THRESHOLD = a(Kp_MB) \times CORR_THR_Fi$
 $= 1.0 \times 40$
 $= 40$

$DET_THRESHOLD = a(Kp_MB) \times DET_THR_Fi$
 $= 1.0 \times 56$
 $= 56$

なるパラメータ $CORR_THRESHOLD$ 、 $DET_THRESHOLD$ がブロック歪判定処理及び補正処理に使用されることになる。また、例えば画像伝送速度が8Mbpsで符号化難易度

Kp_Fi が $4 \leq Kp_Fi < 6$ 、符号化難易度 Kp_MB が $8 \leq Kp_MB < 12$ であるときには、

$CORR_THRESHOLD = a(Kp_MB) \times CORR_THR_Fi$

$= 0.8 \times 2.4$

$= 1.9.2$

$DET_THRESHOLD = a(Kp_MB) \times DET_THR_Fi$

$= 0.8 \times 3.2$

$= 2.5.6$

なるパラメータ $CORR_THRESHOLD$ 、 $DET_THRESHOLD$ がブロック歪判定処理及び補正処理に使用されることになる。

【0154】このようにして、大局的にとらえた画面全体の符号化難易度 Kp_Fi と、画面内の動きをとらえたブロック単位の符号化難易度 Kp_MB を用いて、画像の圧縮率すなわち画像伝送速度に応じて適応的に最適なブロック歪の除去処理を行う。

【0155】

【発明の効果】以上のように、本発明によれば、入力画像データに動き予測符号化とは独自の動き検出処理を行うことにより、検出された動きベクトルと動き補償残差から符号化の難易度を示すパラメータを得るとともに、上記入力画像データからブロック歪判定に必要なパラメータを演算して、ブロック歪を判定し、上記ブロック歪の判定結果に応じて、算出された補正值による補正を上記入力画像データに対して施すことにより、ブロック歪判定の誤判定を軽減するとともにブロック歪除去の効果を改善できる。

【0156】すなわち、例えばMPEG符号化方式とは独自のプリプロセッサによる動き検出工程で算出された動きベクトルと動き補償残差を用いることで、MPEGデコーダとは独立してブロック歪除去を行うことが可能になる。

【0157】これにより、イントラ/インターの区別なくすべての動き検出単位において動き補償残差を得ることができ、パラメータとしての扱いが容易になる。

【0158】また、画像の圧縮率すなわち画像伝送速度に応じてブロック歪判定のための閾値を制御することにより、ブロック歪判定の誤判定を軽減するとともにブロック歪除去の効果を改善することができる。

【図面の簡単な説明】

【図1】本発明に係るブロック歪低減装置の構成例を示すブロック図である。

【図2】上記ブロック歪低減装置におけるプリプロセッサの構成例を示すブロック図である。

【図3】符号化難易度の低い画像と符号化難易度の高い画像の動き補償残差を比較して示す図である。

【図4】符号化難易度の低い画像と符号化難易度の高い画像の動きベクトルの差分値を比較して示す図である。

【図5】上記ブロック歪低減装置によるブロック歪除去処理の手順を示すフローチャートである。

【図6】上記ブロック歪低減装置によるブロック歪

除去の水平方向処理における各処理の位置関係を模式的に示す図である。

【図7】上記ブロック歪低減装置によるブロック歪除去の垂直方向処理における各処理の位置関係を模式的に示す図である。

【図8】上記水平方向処理における符号化難易度係数の算出処理手順を示すフローチャートである。

【図9】上記水平方向処理における垂直相関係数の算出処理手順を示すフローチャートである。

【図10】上記水平方向処理における垂直相関の検出方法を説明するためのDCTブロック境界近傍の画素を示す図である。

【図11】上記垂直相関の検出に用いるエッジ抽出回路の構成を示す図である。

【図12】上記ブロック歪低減装置によるブロック歪除去処理におけるブロック歪判定及び補正処理の具体的な処理手順を示すフローチャートである。

【図13】上記ブロック歪低減装置における水平方向処理のブロック歪除去範囲を示す図である。

【図14】上記ブロック歪低減装置における垂直方向処理のブロック歪除去範囲を示す図である。

【図15】上記ブロック歪低減装置によるブロック歪の除去方式を説明するための図である。

【図16】補正が十分でない補正例を示す図である。

【図17】破綻を生じ易い補正例を示す図である。

【図18】上記ブロック歪低減装置によるブロック歪除去処理における特殊処理1の処理手順を示すフローチャートである。

【図19】補正の結果、新たにエッジを生じる補正例を示す図である。

【図20】修正した補正值による補正例を示す図である。

【図21】ブロック歪がブロック境界に近接する画素上にある場合の補正例を示す図である。

【図22】上記ブロック歪低減装置によるブロック歪除去処理における特殊処理2の処理手順を示すフローチャートである。

【図23】本発明に係るブロック歪低減装置の他の構成例を示すブロック図である。

【図24】上記ブロック歪低減装置によるブロック歪除去処理におけるブロック歪判定及び補正処理の他の処理手順を示すフローチャートである。

【図25】フィールド毎の符号化難易度の算出過程において、静止画検出、シーンチェンジ検出、時間軸フィルタ処理を施す場合の処理手順を示すフローチャートである。

【図26】符号化難易度減衰値の変化を示すタイミングチャートである。

【図27】シーンチェンジ処理後の符号化難易度の変化を示すタイミングチャートである。

【図28】場面単位の符号化難易度からブロック歪判定処理及び補正処理に使用するパラメータを決定するのに使用するテーブルの例を示す図である。

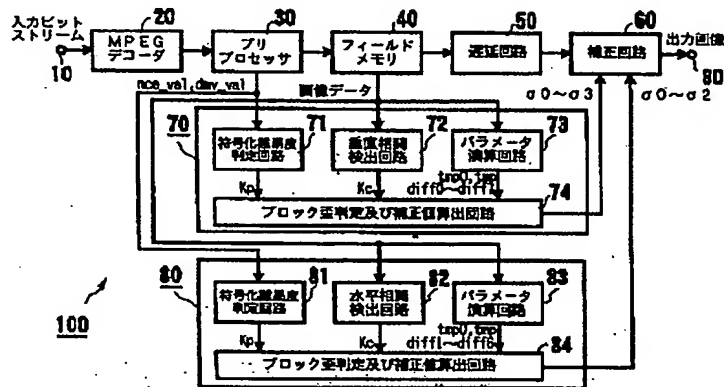
【図29】ブロック単位の符号化難易度から重み付け係数を決定するのに使用するテーブルの例を示す図である。

【符号の説明】

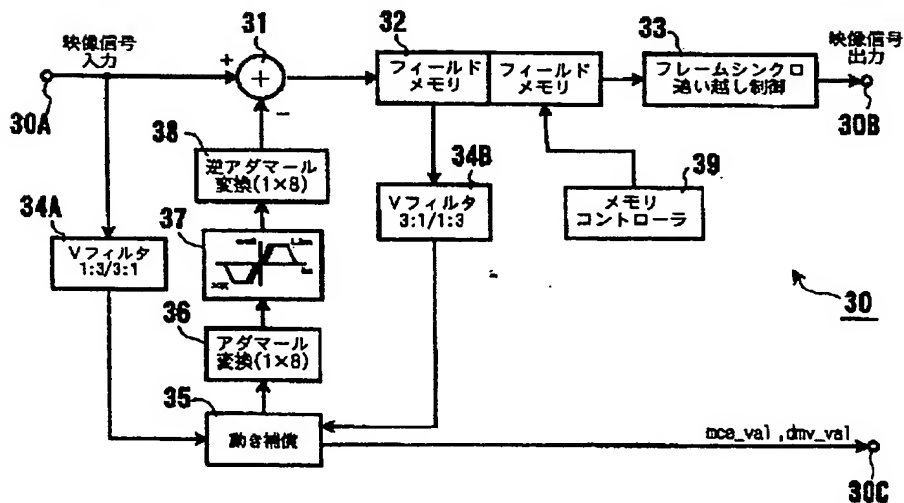
10, 210 入力端子、20, 220 MPEGデコーダ、30, 230 プリプロセッサ、40, 241、2

42 フィールドメモリ、50, 271, 272 遅延回路、60, 281, 282 補正回路、70, 260 水平方向処理部、71, 81 符号化難易度判定回路、72 垂直相関検出回路、73, 83 パラメータ演算回路、74, 84 ブロック歪判定及び補正値算出回路、80, 250 垂直方向処理部、82 水平相関検出回路、90, 290 出力端子、100, 200 ブロック歪低減装置

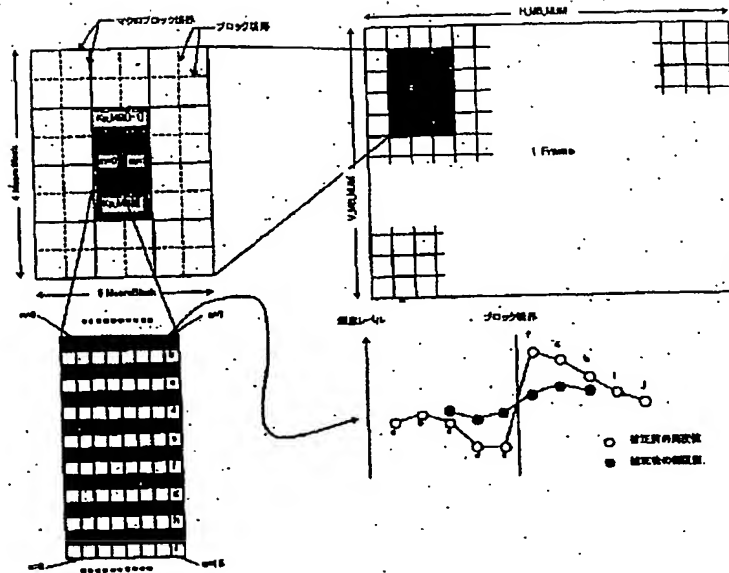
【図1】



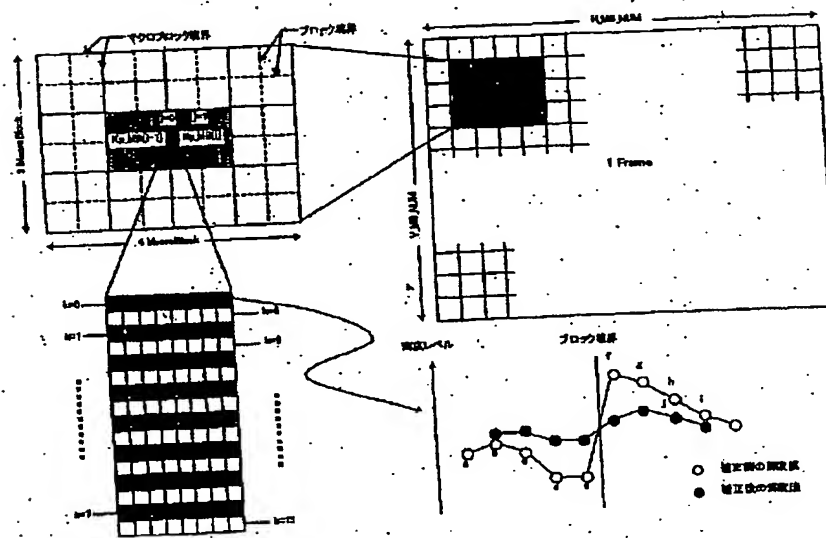
【図2】



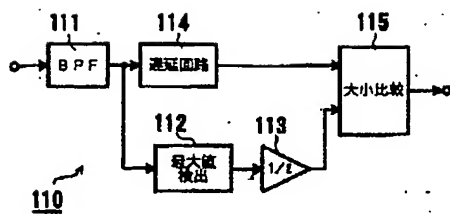
【図 6】



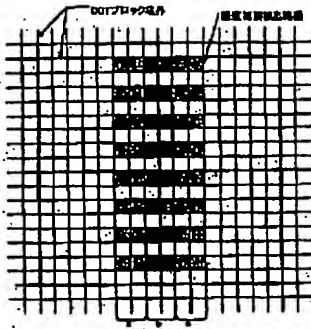
【図 7】



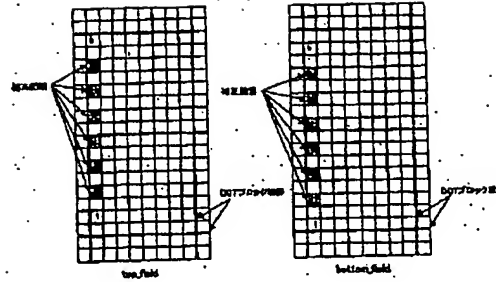
【図 11】



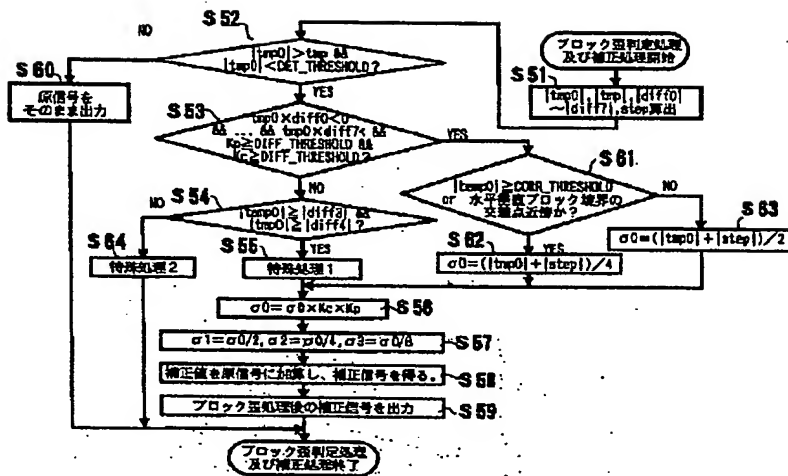
【図10】



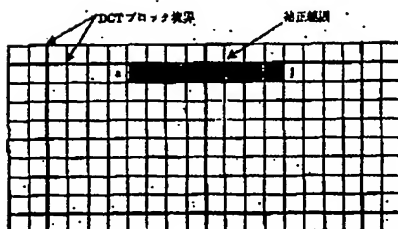
【図14】



【図12】



【図13】



【図29】

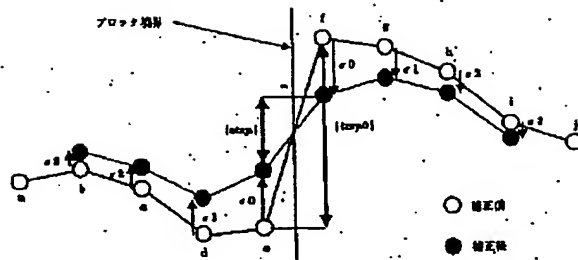
(A) 伝送速度: 4Mbps

Kp_MB	a(Kp_MB)
14	1.5
10	1.0
6	0.8
2	0.4
	0.2

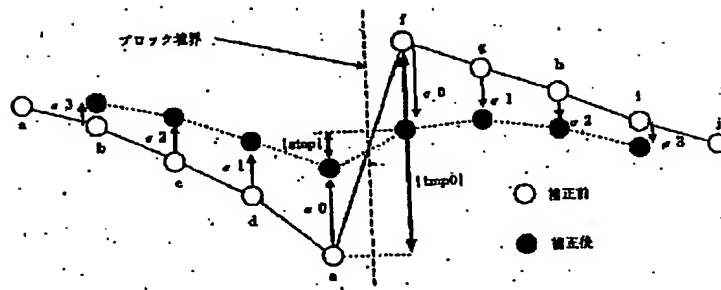
(B) 伝送速度: 8Mbps

Kp_MB	a(Kp_MB)
12	1.0
8	0.8
6	0.6
2	0.4
	0.2

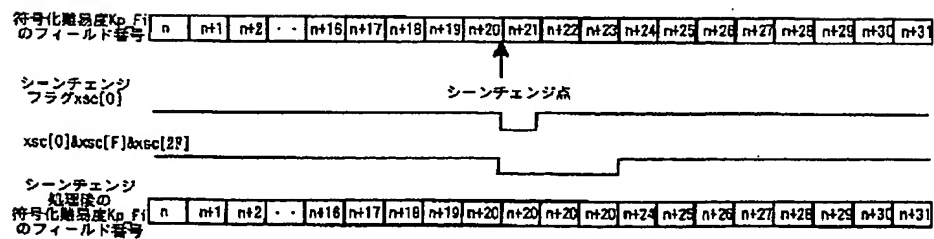
【図15】



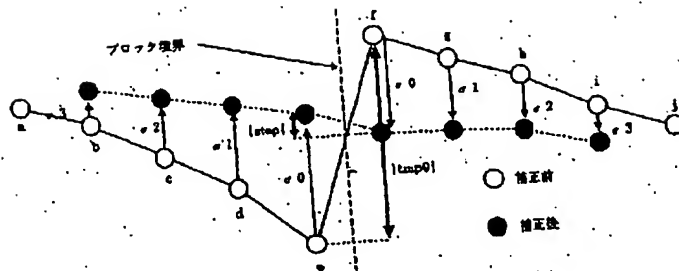
【図16】



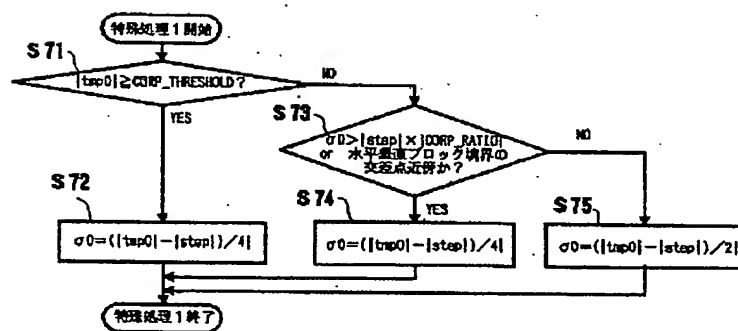
【図27】



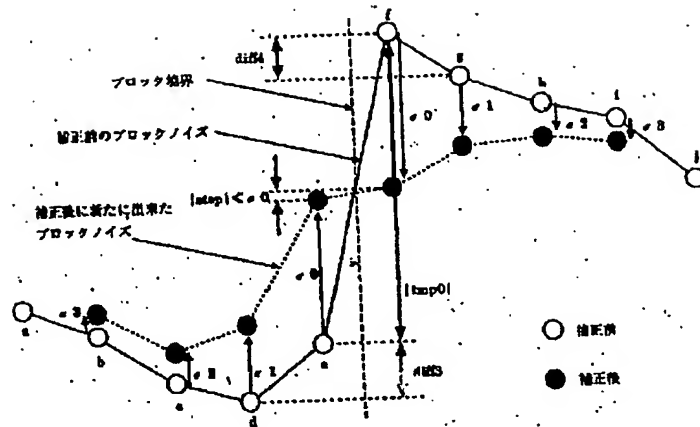
【図17】



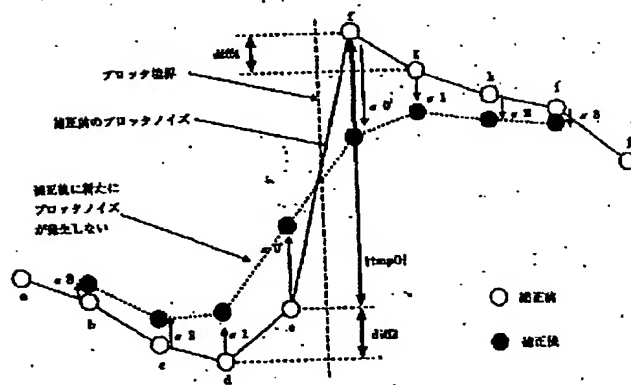
【図18】



【図 19】



【図 20】

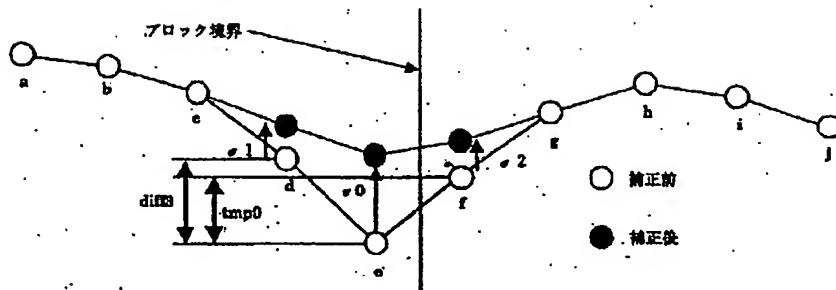


【図 26】

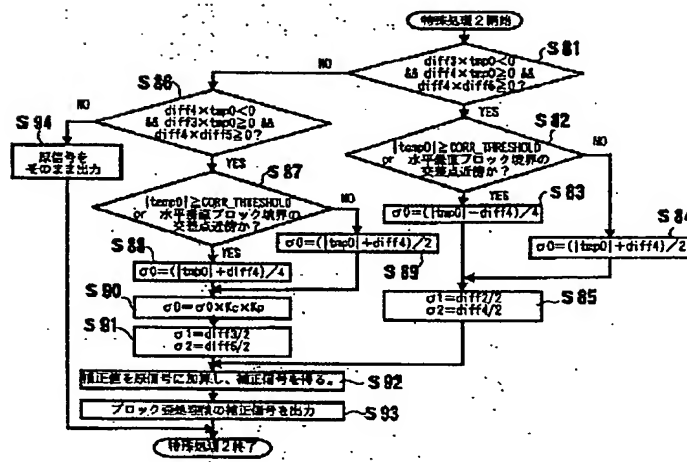
符号化難易度 Kp, Fi のフィールド番号	n	n+1	n+1	...	n+16	n+17	n+18	n+18	n+20	n+21	n+22	n+23	n+24	n+25	n+26	n+27	n+27	n+28	n+30	n+31
	停止画																			

符号化難易度 減衰値 d	0	0	0	...	2	4	6	8	10	12	14	15	13	11	9	7	5	3	1	0
-----------------	---	---	---	-----	---	---	---	---	----	----	----	----	----	----	---	---	---	---	---	---

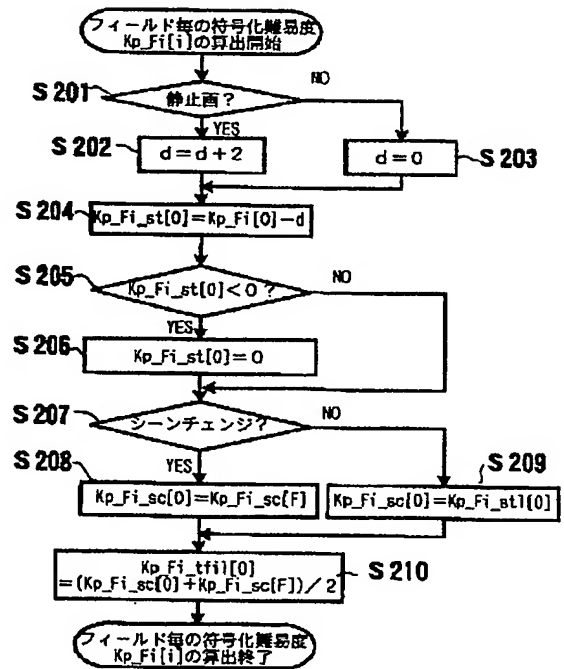
【図21】



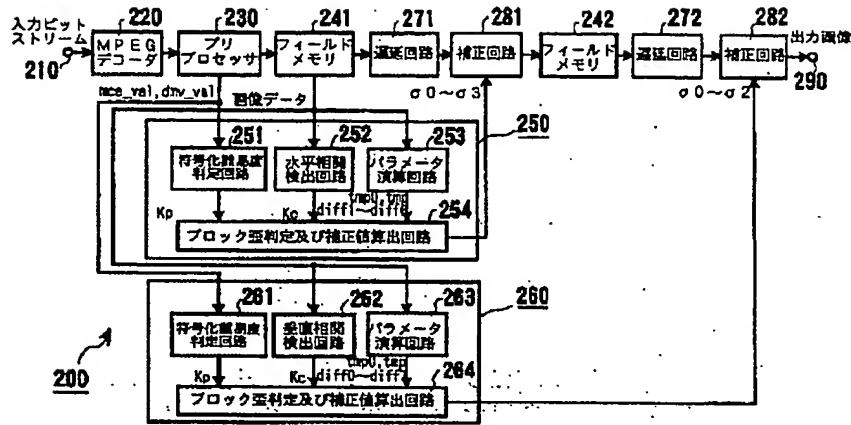
【図22】



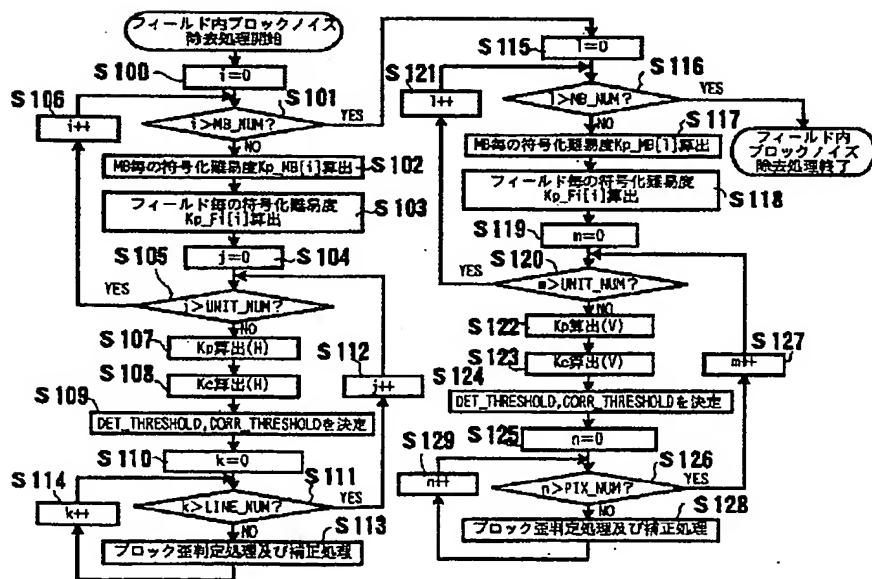
【図25】



【図 23】



【図 24】



【図28】

(A) 伝送速度：4Mbps

Kp_Fi	CORR_THR_Fi	DET_THR_Fi
12	56	72
8	40	56
4	24	32
2	16	24
	8	16

(B) 伝送速度：8Mbps

Kp_Fi	CORR_THR_Fi	DET_THR_Fi
10	40	56
6	32	40
4	24	32
2	8	16
	4	8

【手続補正書】

【提出日】平成12年7月11日（2000. 7. 11）

【手続補正1】

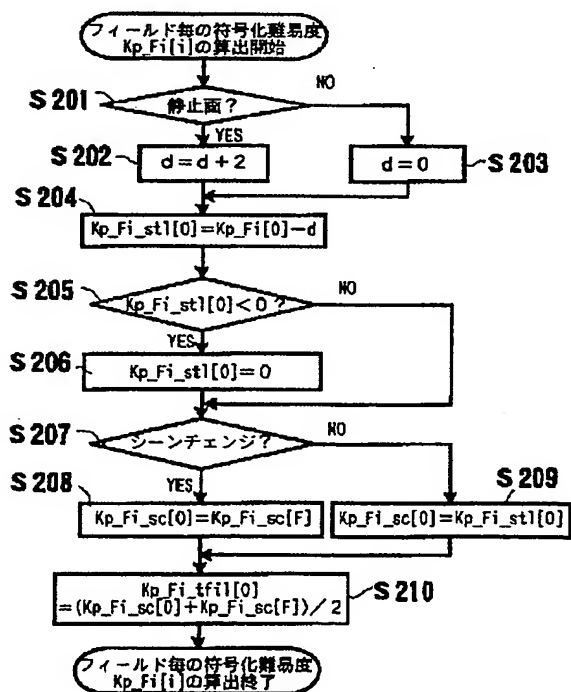
【補正対象書類名】図面

【補正対象項目名】図25

【補正方法】変更

【補正内容】

【図25】



【手続補正2】

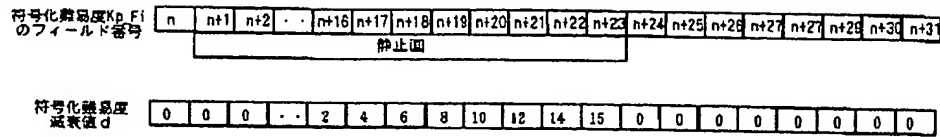
【補正対象書類名】図面

【補正対象項目名】図 2 6

【補正内容】

【補正方法】変更

【図 2 6】



フロントページの続き

(72)発明者 福田 京子
 東京都品川区北品川 6 丁目 7 番 35 号 ソニ
 ー株式会社内

(72)発明者 加藤 元樹
 東京都品川区北品川 6 丁目 7 番 35 号 ソニ
 ー株式会社内

(72)発明者 小幡 功史
 東京都品川区北品川 6 丁目 7 番 35 号 ソニ
 ー株式会社内

F ターム (参考) 5C059 KK03 MA00 MA03 MA05 MA23
 NN01 NN43 PP25 PP26 TA11
 TA68 TB07 TC10 TC11 TC38
 TD08 UA02

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-094996

(43)Date of publication of application : 06.04.2001

(51)Int.Cl.

H04N 7/32

H04N 7/30

(21)Application number : 2000-177646

(71)Applicant : SONY CORP

(22)Date of filing : 13.06.2000

(72)Inventor : IKI SHINYA

KOBAYASHI HIROSHI

FUKUDA KYOKO

OBATA KOJI

KATO MOTOKI

(30)Priority

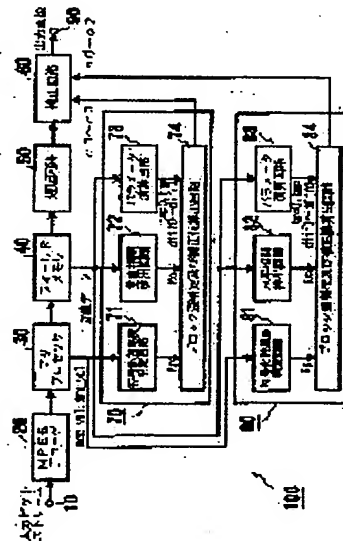
Priority number : 11205258 Priority date : 19.07.1999 Priority country : JP

(54) BLOCK DISTORTION REDUCTION METHOD AND BLOCK DISTORTION REDUCTION DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To stably reduce or eliminate block distortion through simple processings and to prevent errors in discrimination, in the case of discriminating the block distortion.

SOLUTION: Block distortion discrimination and coding difficulty discriminating circuits 74, 84 obtain a parameter denoting the difficulty of coding from a motion vector and a motion compensation residue, which are obtained by applying motion detection processing different from motion prediction coding to received image data by a pre-processor 30, parameter arithmetic circuits 73, 83 calculate a parameter required for block distortion discrimination from the received image data, the block distortion discrimination and coding difficulty discriminating circuits 74, 84 discriminate block distortion, on the basis of the parameter denoting the difficulty of coding and the calculated parameters to calculate a correction value for reducing the block distortion, and a correction circuit 60 subjects the received image data to a correction using the correction value.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

3-2 an English language explanation
of Japanese laid-open patent publication No. 2001-094996

This document describes the method to detect the block boundary where block noise appears based on “a degree of difficulty for coding” and “a calculated parameter”.

3-3 English translations of the surrounded parts by the red line
in Japanese laid-open patent publication No. 2001-094996
([0155] to [0158])

[0155]

[The Effect of the Invention] As described above, according to the present invention, it is possible to obtain a parameter indicating the difficulty of coding an image from a motion vector and a motion compensation residue by applying motion detection processing to the input image data, which processing is different from the motion predictive coding; calculate from the input image data a parameter necessary for discriminating the block deformation, discriminate the block deformation by using the parameter; and compensate the input image data for the calculated compensating value on the basis of the block deformation discrimination result. The above described operation of the device can reduce the possibility of erroneous discrimination of the block deformation and, at the same time, improve the removal of the block deformation.

[0156] Namely, it is possible to remove block noises independently from the MPEG decoder, for example, by using a motion vector and a motion compensation residue determined by the motion detection processing by a preprocessor, which processing is different from the MPEG coding method.

[0157] The above-described processing can determine the motion compensation residue for every motion detection unit without discrimination of intra/inter coding, thereby making it easy to obtain and use the above parameter.

[0158] The threshold for discriminating the block deformation can be controlled in accord with the image compression ratio, i.e., the image transmission rate, resulting in reducing the possibility of erroneous discrimination of the block deformation and, at the same time, improving the effect of removing the block deformation.

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.